(12)

AD-A231 168

Office of Naval Research

Final Report

Programming with Articulated Objects

Contract: N00014-88-K-0632

June 1, 1988 through September 30, 1990

Joseph K. Kearney
Department of Computer Science
University of Iowa
Iowa City, IA 53342

Net Mail Address: kearney@cs.uiowa.edu

Principal Investigator: Joseph K. Kearney
Institution: University of Iowa
Phone Number: (319) 335-0741
Net Mail Address: kearney@cs.uiowa.edu
Contract Number: N00014-88-K-0632
Contract Title: Programming with Articulated Objects
Reporting Period: June 1, 1988 - September 30, 1990

(1) Summary of Technical Results

The support provided under Office of Naval Research Contract Number N00014-88-K-0632, Programming with Articulated Objects, has enabled us conduct important research in programming methods for the design, analysis, and control of mechanical devices. The contract also supported, in part, the development of new techniques for image-based motion analysis. Work supported by the project was organized under four projects. A brief summary of the accomplishments for each project is described below. The attached publications and reports provide more detailed presentations of the results. Additional publications are currently in preparation and will be forwarded upon completion.

Project 1: Programming Mechanical Simulations

Computer simulation promises to have a dramatic influence on the study of mechanical systems and the interaction between humans and machines. The central goal of this project was to study methodologies for controlling complex physical objects in simulation. Our work provides a conceptual framework that integrates the mechanical and control components of a simulation. The programming paradigm that forms the basis of this framework views a mechanical simulator as a multi-level constraint solver. The usefulness of the approach was demonstrated through a series of experiments examining the control of robot locomotion.

Constraint-oriented Simulation

In our work, simulation is cast as a problem of solving a sequence of time-varying dynamic constraints on the attributes of physical objects. Constraints due to mechanical considerations, such as joints between bodies, and constraints due to control considerations, such as guided movements, are treated in a uniform fashion. At any instant in time, a system of linear equations constrains the the instantaneous motions of bodies. These motion equations relate accelerations of bodies to the forces and torques on the bodies. Given the positions and velocities of all bodies at time t, positions and velocities at time $t + \Delta t$ are found by solving the motion equations for accelerations and integrating from previous states. Given initial positions and velocities, this process is iterated to simulate motion over an interval of time.

The set of motion constraint equations is derived from three sources. First, a base set of equations modeling the fundamental laws of motion and the mechanical connections between objects is automatically formulated from a model definition. Secondly, the base set of equations may be reformulated during the course of a simulation to accommodate changing relationships among objects as new contacts are made and old contacts are broken. Lastly, control routines may influence the composition of the set of motion equations. The user conducting the simulation models control constraints by programmatically adding and deleting named equations to the set of motion equations.

In many simulation systems including ADAMS, DADS, and SD/FAST,[2,7,14,16] motion constraints are predetermined by a model of the mechanism and remain unchanged during simulation. By allowing conditional reformulation of motion constraints and by incorporating control constraints into the set of motion equations, the power and expressiveness of simulation programs is significantly extended. However, the constraint restructuring must be accomplished in a carefully disciplined manner to avoid comprising the integrity of the simulation. We have developed a regimen under which the flexibility of constraint editing is achieved without disrupting the basic integration process.

Object-level Programming for Locomotion

We demonstrated our control methods by simulating robot locomotion. Using the constraint-based programming paradigm, we derived model-independent control algorithms for hopping. We have successfully simulated hopping for a variety of designs. Our approach concentrates on the interaction between the object and its surroundings

at points of contact. It is only through reactions at contact points that a hopper can alter it momenta. Control programs are expressed as dynamic constraints on the external forces and torques applied to the hopper by pushing against surfaces in contact.

The nature of contact relations places strict constraints on the dynamics of control problems. Control is organized as a two-step process. Within the bounds of contact constraints, a plan for applying forces and torques to the whole hopper is devised. The use of a generalized, virtual leg permits programs to express the motion of hopping bodies independent of their structure. Secondly, the controllable degrees of freedom are used to achieve the required contact forces.

The resulting programs are comparable to *object-level* programs for robot manipulation. An object-level language specifies robot operations by defining the desired state of the object to be manipulated.[3, 22, 23, 29, 30, 32] The robot actions required to bring about the necessary changes to the object are determined by lower levels of the robot system. In a similar way, contact constraints treat the robot as an object to be pushed or spun. The actuator values required to achieve a desired external force or torque are determined by lower level processes.

To demonstrate the generality of our approach, we've tested a variety of hoppers using the same high-level control program. The two basic designs we've used are shown in Figure 1. The hopper on the left models the CMU one-legged hopper.[31] The hopper on the right emulates the torso, upper leg, and lower leg of an anthropoid. A single hop by the anthropomorphic hopper is shown in Figure 2. Both hoppers were robustly controlled by the same set of contact constraints.

The usefulness of contact constraint programming goes beyond basic hopping. A simple, one-legged robot, such as the CMU hopper, has contact with the ground during stance at a single point. The point contact permits only a single constraint force to be applied to the hopper. For this reason, the external force and moment on the hopper are intrinsically coupled. More complex interactions afford a rich set of alternatives for control required to achieve many behaviors. For example, a hopper with a sizable foot touching the ground shares a plane of contact with the support surface. With planar contact, the hopper can independently control the external force and moment during stance. We've demonstrated the increased controllability by simulating a flip with no change in horizontal velocity. Because the external force and moment are decoupled, the hopper can jump vertically with sufficient momentum to rotate through a complete circle. We believe that the contact constraint analysis will prove useful in understanding a wide variety of self-movement tasks.

Technical Reports

J.K. Kearney and D.H. Lee, "Motion control with process abstraction," *Third IEEE International Symposium on Intelligent Control*, Aug. 1988.

J.K. Kearney and S. Hansen, "Generalizing the Hop: Object-level programming for legged motion" submitted to *The International Journal of Robotics Research*, also published as University of Iowa Tech Report UI-CS 90-09

Project 2: Stream Editing for Animation

The first step in creating a computer animation is often the definition of a time-varying geometric model. Powerful motion generation tools enable designers to create physically realistic and biologically plausible object motions through dynamic simulation and constraint-based optimization.[4-6, 8, 15, 24, 25, 33] Motion models can also be determined from measurements of physical motions using techniques of cinematography, photogrammetry, and accelerometry. This project investigated methods for combining and modifying motion sequences and for integrating viewing models with object movements.

Our approach is based on the representation of movement sequences as streams. We have developed software that allows streams to be filtered, duplicated, transformed, and combined. Cameras with prescribed motions can be introduced to visualize scenes. A general facility for composing sequences of spatial transforms permits the specification of complex trajectories and relative motion.

The resulting motion sequences can be rendered as animation or presented as the input circumstances for simulation. Man-in-the-loop simulation offers enormous potential for testing and training of human operators in hazardous situations such as flying, driving, and teleoperation of robots. The synthesis of realistic situations is critical for the effective development of computer simulated training and testing applications. Streams provide a conceptual framework for composing complex dynamic settings and a computational model that is robust, reliable, and simple to implement.

We present a simple example of stream manipulation to illustrate how a single stream acquired from a physically-based simulation of walking can be duplicated, transformed, and assembled into a new stream that represents a precision drill team marching in formation. Our starting point is a stream that models a walking figure. The stream was produced by the physical simulator **newton**. To produce a second walking figure whose path is displaced from the first, we create a transformed copy of the walking stream. The two streams are merged to create a stream with a pair of figures, walking in step with one another. The pair of walking figures can be transformed and merged to obtain a quartet of walkers. This process can be repeated a number of times to produce a whole brigade. The brigade is represented as a single stream that can be reproduced, reoriented, repositioned, and combined with the original marchers to create two divisions marching in different directions. With a judicious selection of the transformation parameters, we can have the two formations cross paths, with members of each brigade passing between members of the other.

A scene from the motion sequence with two groups of four walkers is pictured in Figure 3. The operations required to assemble the this model are schematically shown in Figure 4. Edges indicate data streams and boxes indicate stream producing operations. One of our long range goals is to build a graphic programming language similar to these schematic diagrams.

The marching example illustrates the simplicity and power of animation stream editing. The marching example can be constructed even more compactly than presented above by using transform generators that allow sequences of transforms to be

defined. Stream operations focus attention on the flow of data through computational processes. Successive members of the stream funnel through filters or functions that are mapped over the stream. As a consequence, graph representations such as the one shown in Figure 4 are well suited for visualizing networks of stream operations.

The focus of our work has been on the motion of bodies with rigid members and cameras. However, the approach could easily be adapted to include other types of objects that undergo parametric variation over time. Moving light sources with changing brightness or chroma could be added. Deformable objects which bend or grow could also be included.

Technical Reports

J.K. Kearney and S. Hansen, "Stream Editing for Animation" submitted to *The Visual Computer*, also published as University of Iowa Tech Report UI-CS 90-08.

Project 3: Optimal Motion Analysis

This project investigated integrated environments for optimal motion analysis. Optimal motion analysis is of interest for the development of efficient control algorithms for robots,[17] to maximize athletic performance, and to improve rehabilitory treatment.[21] Optimal motion generation has also been proposed as a means to create realistic animations of complex movements.[33] A large number of studies have observed stereotypic motion patterns among humans.[21] It has long been conjectured that the motions naturally preferred by humans for walking, throwing, jumping, and running are minimal with respect to some objective function. Biologists still debate over what the underlying objective function or functions might be. Optimal motion analysis provides a constructive means to test these hypotheses.

A model-driven system for motion optimization named *Optimizer* was completed in the last year of the contract. Optimizer allows a user to simply define planar chains of rigid links. The desired motion of an object is specified by imposing constraints on the state of object. At the user's request, the system will determine a trajectory that satisfies these constraints and is optimal with respect to some objective criteria. The time-varying values of important state variables can be conveniently inspected and results can be recorded for later study and comparison.

Optimizer presents a uniform view of the system to the user. A common language interface is used to define models, to specify constraints, and to inspect experimental results. In addition to the language interface, menus provide a convenient means to navigate through the various design functions. The menu interface provides a user friendly environment, and the user can exploit the power of expressions provided by the language interface to work with the system.

Optimizer solves for optimal motions by discretizing a continuous motion into a sequence of time-varying positions. The granularity of the discretization can be selected by the user. Given an initial sequence, Optimizer formulates a nonlinear least squares problem which is solved with a version of the Levenberg-Marquardt algorithm.

The principal body of Optimizer was programmed in Lisp. Development was facilitated by incorporating a diverse set of software packages to perform many of the required support functions. The special purpose application languages for object definition, constraint specification, and report generation were parsed using the UNIX utilities lex and yacc. User programs were translated into Lisp functions and executed. Symbolic expressions for the equations of motions were derived from the object model using the symbolic mathematics package Macsyma.[1] The motion equations derived in Macsyma and motion constraints specified by the user were combined and translated into a Fortran function, fcn(), that encapsulated the objective function. Given values for all joint positions not specified by the problem constraints, this function returned the torque at each joint at each time step. The set of unconstrained joint positions that minimized sum of squared joint torques was determined with a robust Fortran routine lmdif1() from the MINPACK subroutine package.[11] Graphics routines were borrowed from the simulator Newton.[8] Optimizer demonstrates the range of support functions required for integrated design environments and the importance of incorporating

softwared packages for complex engineering applications.

To illustrate the use of Optimizer, we present the results of an analysis of the swinging motion of a golfer. The model consists of two links. One link represents the golfer's arms and the second link represents the golf club. The link representing the arms is constrained to rotate about a fixed axis located at the shoulder. The motion is initiated with the joints positioned to begin the forward swing of the club and with no angular velocity. The final configuration constrains the club head to be in contact with a ball. The final joint velocities are specified such that the club head has a horizontal velocity while striking the ball. Figure 5 illustrates the solution of the golf swing optimization. Link positions are overlayed in the window located upper-left side of the screen. Time-varying values of state variables are displayed in the graphs on the right side of the screen. The motion sequence qualitatively corresponds to motions reported in experiments with real golfers.[10, 19]

Technical Reports

B. Prasad, "Optimizer: A Model Driven System for the Study of Optimal Motions," MA thesis, University of Iowa, August, 1990.

Project 4: Image Correspondence for Particles

Accurate measurement of particle location and motion is important for a diverse assortment of scientific and engineering problems including the design of ship hulls, automobile cylinders, and heart valve replacements. A particle is a small, compact object such as an oil droplet, a blood cell, or a piece of dust. Fluid motions can be quantitatively assessed by tracking particles seeded in flow fields. The three-dimensional distribution of particles at an instant in time can also give important information about the characteristics of gas and fluid motions. This project examined the problem of measuring the location of large numbers of particles at a snapshot in time using image processing techniques. The algorithm we devised can be simply extended to measure particle motions.

Information about the three-dimensional location of an object can be recovered from multiple images of the object taken simultaneously from different viewpoints.[9] This process is usually performed in three steps. First, sets of object features are independently found in each image. An object feature is a visually distinctive characteristic that can be reliably detected from different viewpoints. The edges and corners of surface markings are often used as features. Next, features are matched across the images. The last step is to estimate the three-dimensional location of the feature by triangulation. The use of two or more views to infer three-dimensional structure and shape is called stereo vision. Considerable research has been directed towards methods that rely on two cameras. Recently, a number of papers have pointed out the advantages of introducing a third camera.[12, 13, 18, 20, 26-28, 34-36]

This project addressed the matching problem for particles. The particles to be viewed were assumed to be small spheres with a size near the resolution limit of the camera. For our model, we assumed that particles appeared as points in an image. Our problem was to find points in two or more different images that were projections of the same particle. This presents a difficult matching problem, because all image points look alike. Further, neighboring image points may lie at disparate depths. As a consequence, stereo matching algorithms that rely on surface continuity or the appearance of distinctive markings are inappropriate for particle matching.

We devised a general algorithm for matching image points in three views. The method relies solely on geometric constraints based on the camera arrangement. In previous formulations of the trinocular matching algorithm, one image was used to verify the consistency of possible matches between points in the other two images. Our approach generalizes this strategy by considering all pairwise matches between points in all images. We formulate the matching problem as a search for edges contained in all perfect matchings of a bipartite graph. The bipartite graph links points between two images that satisfy a geometric consistency test. To take maximal advantage of the geometric constraints, three such problems must be solved. Our algorithm employs an iterative search to find certain matches. The algorithm has been tested with synthetically generated particle images. The results show that it finds significantly greater numbers of matches than earlier methods when matching uncertainty is high.

An attractive property of our algorithm is its graceful degradation in response to image distortion and modeling error. Although measurement errors may prevent successful matching, wrong matches can almost always be avoided if the error in the image position of a particle can be bounded. Thus, noise can cause a loss of acuity but should not cause the introduction of gross misinformation that could result from incorrect matching. The density of matches obtained by the method can be optimized by regulating the number of particles seeded in the volume and by selectively arranging the cameras to minimize ambiguity in the matching process. The method can be simply extended to match image streaks formed by using long exposure views of moving particles.

Technical Reports

J.K. Kearney, X. Yang, and S. Zhang, "Camera calibration using geometric constraints," *IEEE Conference on Computer Vision* and Pattern Recognition, June 1989, pp. 672-679.

J.K. Kearney "Trinocular Correspondence for Particles and Streaks," submitted to the *IEEE Conference on Computer Vision and Pattern Recognition.*

References

1. "Macsyma User's Guide," SM10500040.0, Symbolics, Inc., January, 1988.
2. *DADS User's Manual*, Computer Aided Design Software, Inc., Oakdale, IA, 1989
3. A. Ambler, S. A. Cameron, and D. F. Corner, "Augmenting the RAPT robot language," in *Languages for Sensor-based Control in Robotics*, ed. Hormann, pp. 305-316, Springer-Verlag, New York, NY, 1987.
4. W. W. Armstrong and M. W. Green, "The dynamics of articulated rigid bodies for purposes of animation," *The Visual Computer*, vol. 1, pp. 231-240, 1985.
5. D. Bae and E.J. Haug, "A recursive formulation for constrained mechanical system dynamics: Part I, open loop systems," *to appear in Mechanics of Structures and Machines*.
6. D. Bae and E.J. Haug, "A recursive formulation for constrained mechanical system dynamics: Part II, closed loop systems," *to appear in Mechanics of Structures and Machines*.
7. M. Chace, "Methods and experience in computer aided design of large displacement mechanical systems," in *Computer Aided Analysis and Optimization of Mechanical System Dynamics*, ed. E.J. Haug, vol. 9, NATO ASI Series F: Computer and System Sciences, Springer-Verlag, 1984.
8. J. Cremer, "An Architecture for General Purpose Physical Simulation -- Integrating Geometry, Dynamics, and Control," Ph.D. Thesis, TR 89-987, Cornell University, April, 1989.
9. J.R. Dhond and J.K. Aggarwal, "Structure from stereo - A review," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19, no. 6, pp. 1489-1510, November 1989.
10. G. Dyson, in *The Mechanics of Athletics*, Meier Publishers, 1980.
11. B. S. Garbow and K. E. Hillstrom, "User Guide for Minpack--1," ANL-80-74, Argonne National Laboratory, Argonne, Illinois, August 1980.
12. A. Gerhard, H. Platzer, J. Steurer, and R. Lenz, "Depth extraction by stereo triples and a fast correspondence estimation algorithm," *8th International Conference on Pattern Recognition*, p. 512, October 1986.
13. E. Gurewitz, I. Dinstein, and B. Sarusi, "More on the benefit of a third eye for machine stereo perception," *Proceedings of the Eighth International Conference on Pattern Recognition*, pp. 966-968, October 1986.
14. E.J. Haug, in *Computer Aided Kinematics and Dynamics of Mechanical Systems*, vol. 1, Allyn and Bacon, Boston, MA, 1989.
15. C. M. Hoffmann and J. E. Hopcroft, "Simulation of physical systems from geometric models," *IEEE Journal of Robotics and Automation*, vol. RA-3, no. 3, pp. 194-206, June 1987.
16. M.G. Hollars, D.E. Rosenthal, and M. Sherman, *SD/FAST User's Manual*, Symbolic Dynamics, Inc., Mountain View, CA, 1990.

17. J.M. Hollerbach and K.C. Suh, "Redundancy Resolution of Manipulators through Torque Optimization," *IEEE Journal of Robotics and Automation*, vol. RA--3, no. 4, pp. 308-316, August 1987.

18. M. Ito and A. Ishii, "Range and shape measurement using three-view stereo analysis," *Proceedings Computer Vision and Pattern Recognition*, p. 9, 1986.

19. T. Jorgensen, Jr., "On the Dynamics of the Swing of a Golf Club," *American Journal of Physics*, vol. 38, no. 5, pp. 644-651, May 1970.

20. Y. Kitamura, K. Masatoshi, and M. Yachida, "Trinocular vision using edge continuity and local features," *Systems and Computers in Japan*, vol. 19, no. 12, pp. 12-23, December 1988.

21. E. Kreighbaum and K. Barthels, *Biomechanics: A Qualitative Approach for Studying Human Movement*, Macmillan Publishing Company, New York, NY, 1990.

22. J. C. Latombe, C. Laugier, J. M. Lefebvre, E. Mazer, and J. F. Miribel, "The LM robot programming system," in *Robotics Research - The Second International Symposium*, ed. H. Inoue, pp. 377-391, MIT Press, Cambridge, MA, 1985.

23. T. Lozano-Perez, "Robot Programming," *Proceedings of the IEEE*, vol. 71, no. 7, July, 1983.

24. N. Magnenat-Thalmann and D. Thalmann, *Computer Animation: Theory and Practice*, Springer-Verlag, Tokyo, Japan, 1985.

25. N. Magnenat-Thalmann and D. Thalmann, *State-of-the-art in Computer Animation: Proceedings of Computer Animation '89*, Springer-Verlag, Tokyo, Japan, 1989.

26. V. J. Milenkovic and T. Kanade, "Trinocular vision using photometric and edge orientation constraints," *Proceedings of the Image Understanding Workshop*, p. 163, December 1985.

27. Y. Ohta, T. Yamaoto, and K. Ikeda, "Collinear trinocular stereo using two-level dynamic programming," *Proceedings of the Ninth International Conference on Pattern Recognition*, pp. 658-662, November 1988.

28. M. Pietikainen and D. Harwood, "Depth from three camera stereo," *Proceedings Computer Vision and Pattern Recognition*, p. 2, 1986.

29. R. Popplestone, A Ambler, and I. Bellos, "An interpreter for a language for describing assemblies," *Artificial Intelligence*, vol. 14, pp. 79-107, 1980.

30. R. Popplestone and A. Ambler, "A language for specifying robot manipulations," in *Robotic Technology*, ed. A. Pugh, pp. 125-141, Peter Peregrinus, London, 1983.

31. M. H. Raibert, *Legged Robots That Balance*, The MIT Press, Cambridge, MA, 1986.

32. R. Volz, "Report of the robot programming language working group: NATO workshop on robot programming languages," *IEEE Journal of Robotics and Automation*, vol. 4, no. 1, pp. 86-90, Feb 1988.

33. Witkin and Kass, "Spacetime Constraints," *SIGGRAPH*, p. 159, 1988.
34. M. Yachida, U. Kitamura, and M. Kimachi, "Trinocular vision: New approach for correspondence problem," *8th International Conference on Pattern Recognition*, p. 1041, October 1986.
35. M. Yachida, "3-D data acquisition by multiple views," *Robotics Research: Third International Symposium*, pp. 11-18, MIT Press, Cambridge, MA, 1986.
36. S. Yokoi, G. Medioni, and R. Nevatia, "Segment-based trinocular vision," *Proceedings of the USA-JAPAN Symposium on Flexible Automation - Crossing Bridges: Advances in Flexible Automation and Robotics*, pp. 817-821.

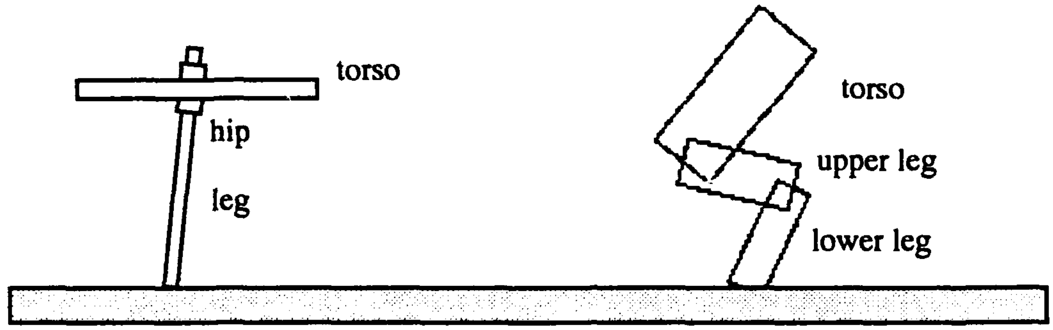

Figure 1. Two hopper designs. The model on the left is an idealization of the CMU one-legged hopper. The model on the right empulates an anthropoid.

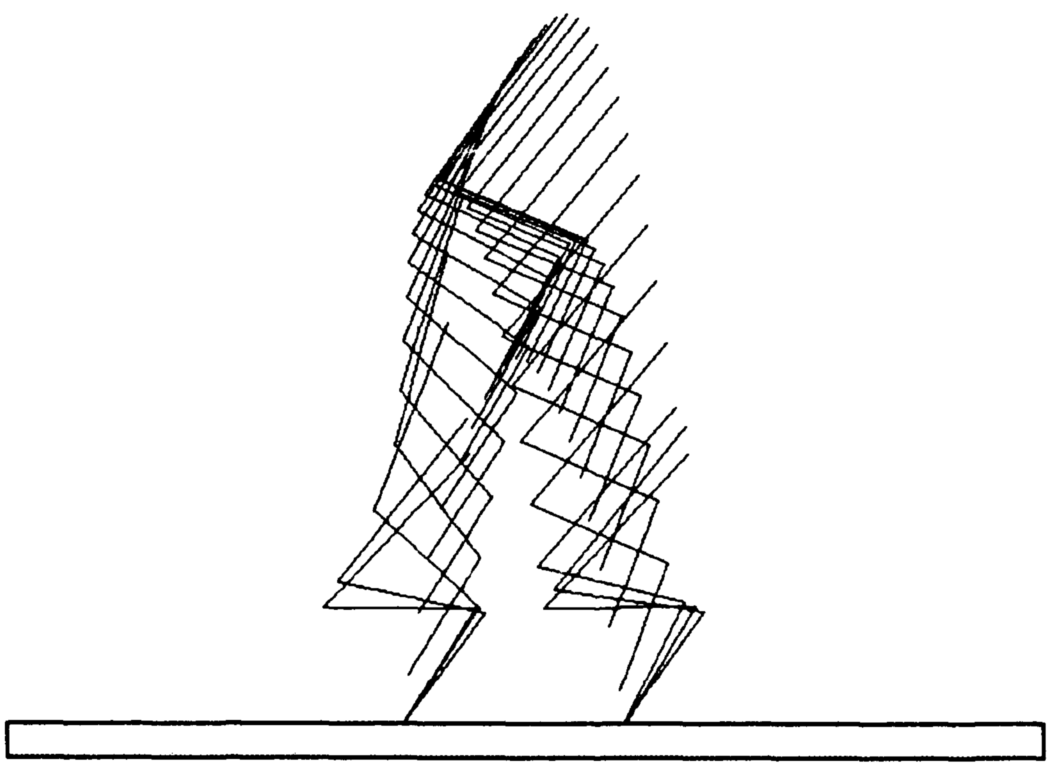

Figure 2. An overlayed sequence of frames from a single hop with the anthropomorphic hopper. The longitudinal axis of each link is drawn.

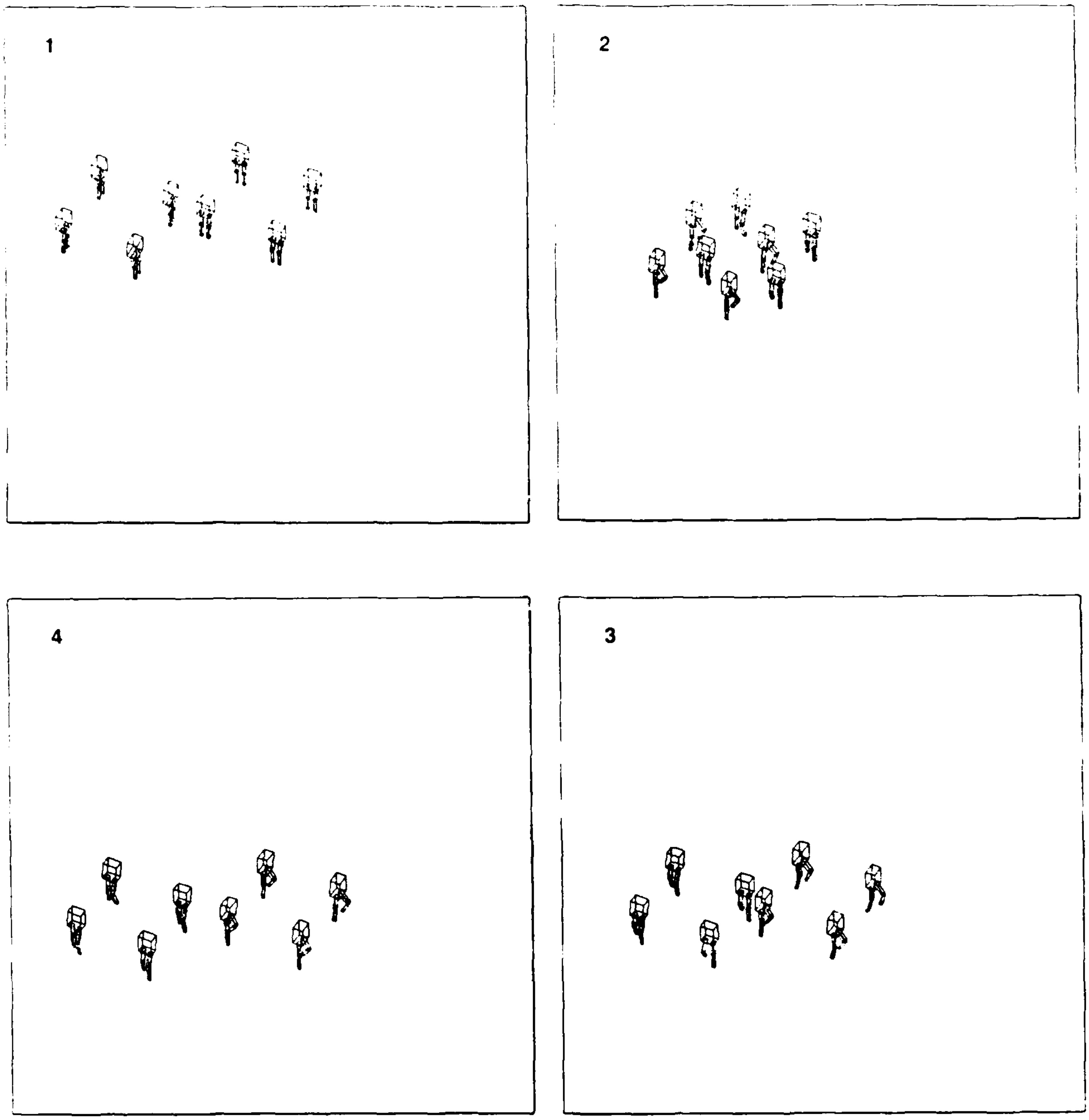

Figure 3. Selected scenes from a motion sequence with eight walking figures moving in groups of four. The scenes are ordered clockwise beginning with the upper, left panel. The sequence was created using the stream editing program diagrammed in Figure 4.

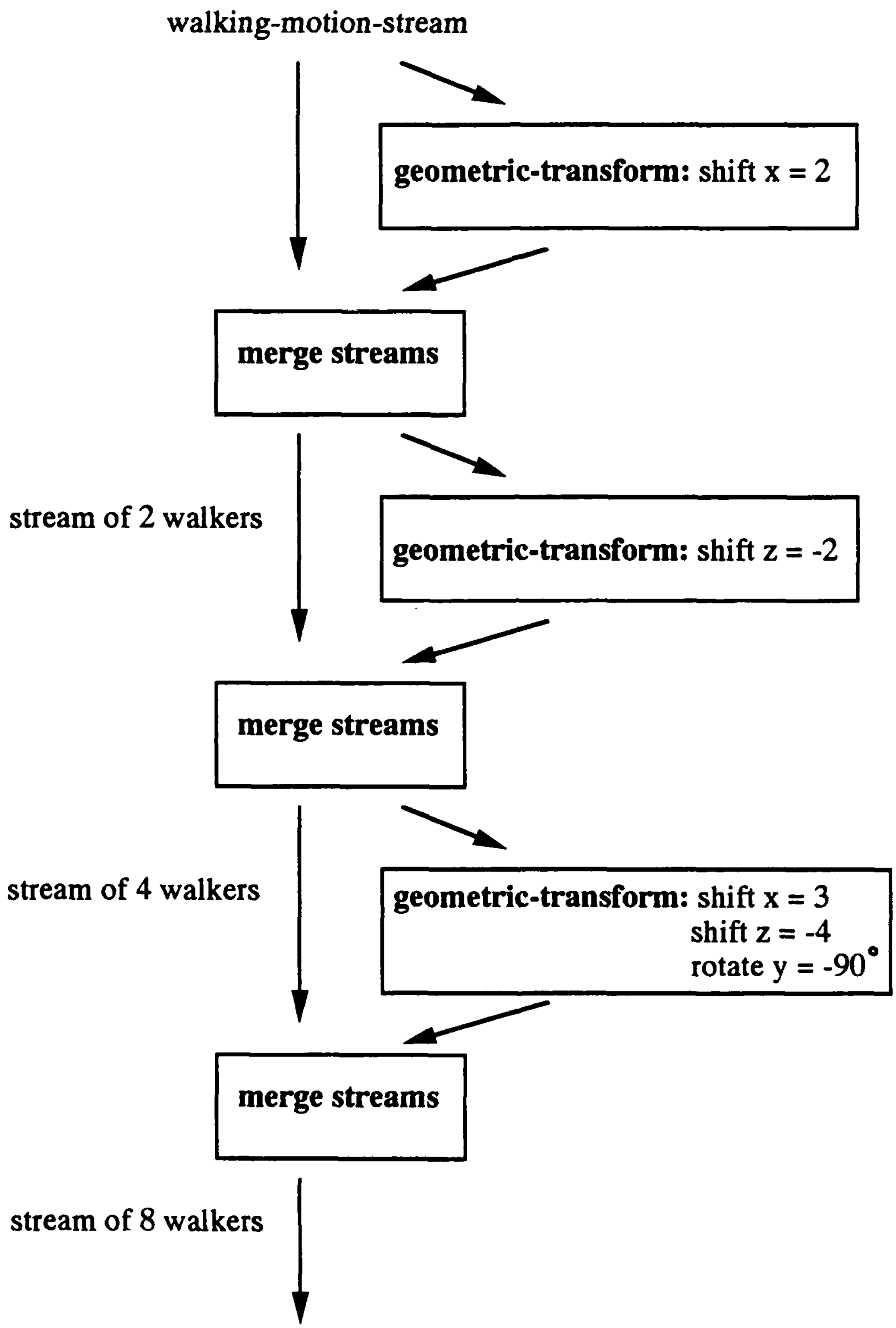

Figure 4. A schematic representation of the stream editting program to create 8 marchers.

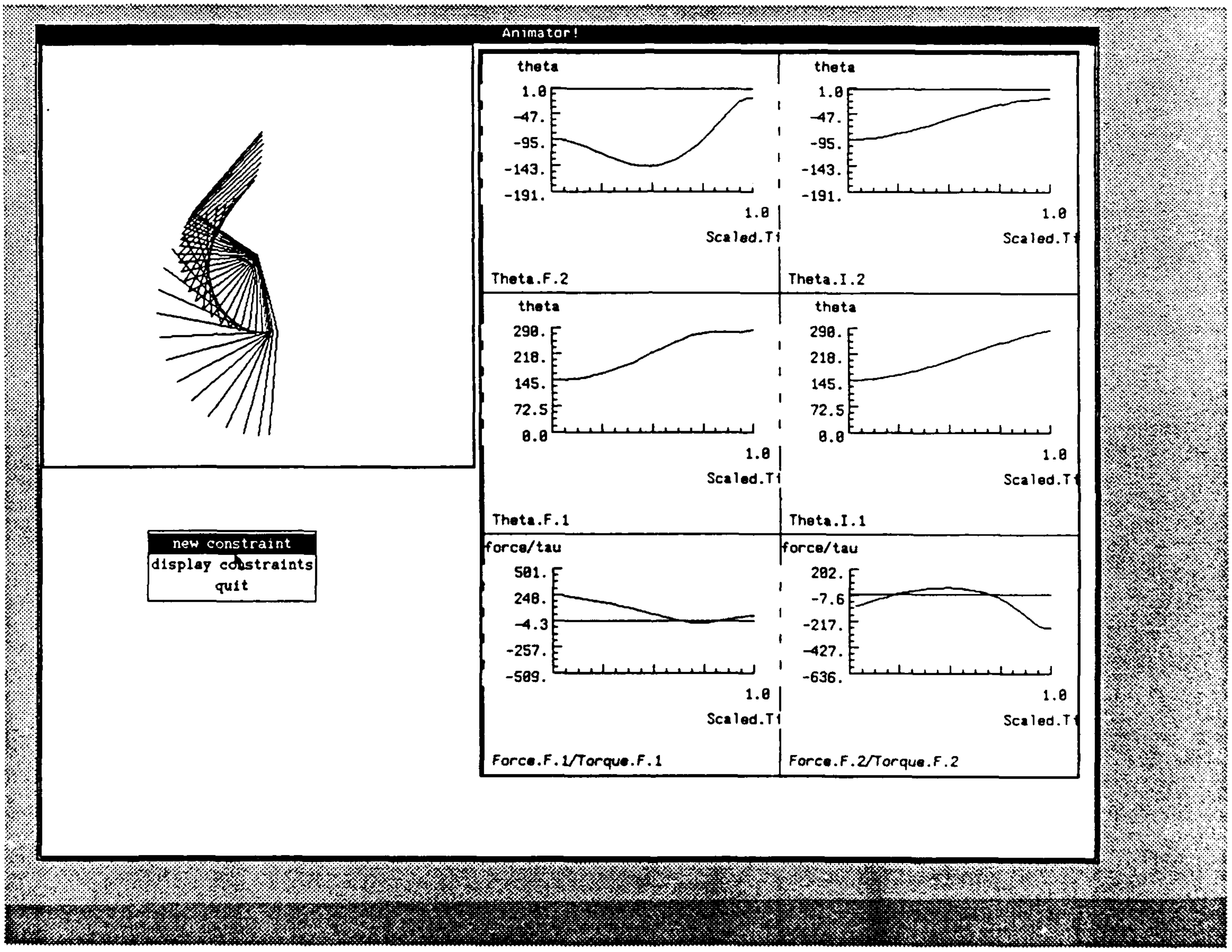

Figure 5. The optimal motion for a two-link, planar manipulator. The linkage was constrained to have zero initial velocity. Final joint velocities were constrained so that tip moved horizontally. Graphs of motion parameters appear on the right. The top four panels shown plot of joint angles over time. Initial values are shown on the right; final values are shown on the left. The bottom two panels give the torque graphs for the estimated optimal motion.

Principal Investigator: Joseph K. Kearney
Institution: University of Iowa
Phone Number: (319) 335-0741
Net Mail Address: kearney@cs.uiowa.edu
Contract Number: N00014-88-K-0632
Contract Title: Programming with Articulated Objects
Reporting Period: June 1, 1988 - September 30, 1990

(2) Productivity Measures

(a) Refereed papers submitted but not yet published: 3
(b) Refereed papers published: 3
(c) Unrefereed reports and articles: 1
(d) Books or parts thereof submitted but not yet published: 0
(e) Books or parts thereof published: 0
(f) Patents filed but not yet granted: 0
(g) Patents granted: 0
(h) Invited presentations: 2
(i) Contributed presentations: 0
(j) Honors received: 5
(k) Prizes or awards received: 0
(l) Promotions obtained: 0
(m) Graduate students supported: 6
(n) Post-docs supported: 0
(o) Minorities supported: 5

Principal Investigator: Joseph K. Kearney
Institution: University of Iowa
Phone Number: (319) 335-0741
Net Mail Address: kearney@cs.uiowa.edu
Contract Number: N00014-88-K-0632
Contract Title: Programming with Articulated Objects
Reporting Period: June 1, 1988 - September 30, 1990

(3) Publications, Presentations and Reports

(a) Articles

(1) J.K. Kearney and D.H. Lee, "Motion control with process abstraction," *Third IEEE International Symposium on Intelligent Control*, Aug. 1988.

(2) J.K. Kearney, X. Yang, and S. Zhang, "Camera calibration using geometric constraints," *IEEE Conference on Computer Vision* and Pattern Recognition, June 1989, pp. 672-679.

(3) J. Hopcroft, J.K. Kearney, and D.B. Krafft, "A case study of flexible object manipulation" to appear in *The International Journal of Robotics Research.*

(4) J.K. Kearney and S. Hansen, "Stream Editing for Animation" submitted to *The Visual Computer*, also published as University of Iowa Tech Report UI-CS 90-08.

(5) J.K. Kearney and S. Hansen, "Generalizing the Hop: Object-level programming for legged motion" submitted to *The International Journal of Robotics Research*, also published as University of Iowa Tech Report UI-CS 90-09.

(6) J.K. Kearney "Trinocular Correspondence for Particles and Streaks," submitted to the *IEEE Conference on Computer Vision and Pattern Recognition.*

(7) B. Prasad, "Optimizer: A Model Driven System for the Study of Optimal Motions," MA thesis, University of Iowa, August, 1990.

(b) Invited Presentations

(1) "Programming with Complex Physical Objects," University of Rome, Rome, Italy, July, 1988.

(2) "Motion Analysis in Time-varying Imagery," University of Rome, Rome, Italy, July, 1988.

Principal Investigator: Joseph K. Kearney
Institution: University of Iowa
Phone Number: (319) 335-0741
Net Mail Address: kearney@cs.uiowa.edu
Contract Number: N00014-88-K-0632
Contract Title: Programming with Articulated Objects
Reporting Period: June 1, 1988 - September 30, 1990

(4) Graduate Students supported under the Contract:

Year of Support	Name
1988	Dong Ho Lee (Ph.D. in progress)
1988	Xiaoli Yang (earned M.S.)
1988	Shenzhi Zhang (earned M.S.)
1989	Angshuman Guha
1989-1990	Bevra Prasad (earned M.S.)
1990	Stuart Hansen